Using Experiments to Build a Body of Knowledge

Victor R. Basili

Experimental Software Engineering Group
Institute for Advanced Computer Studies
Department of Computer Science
University of Maryland
and
Fraunhofer Center - Maryland

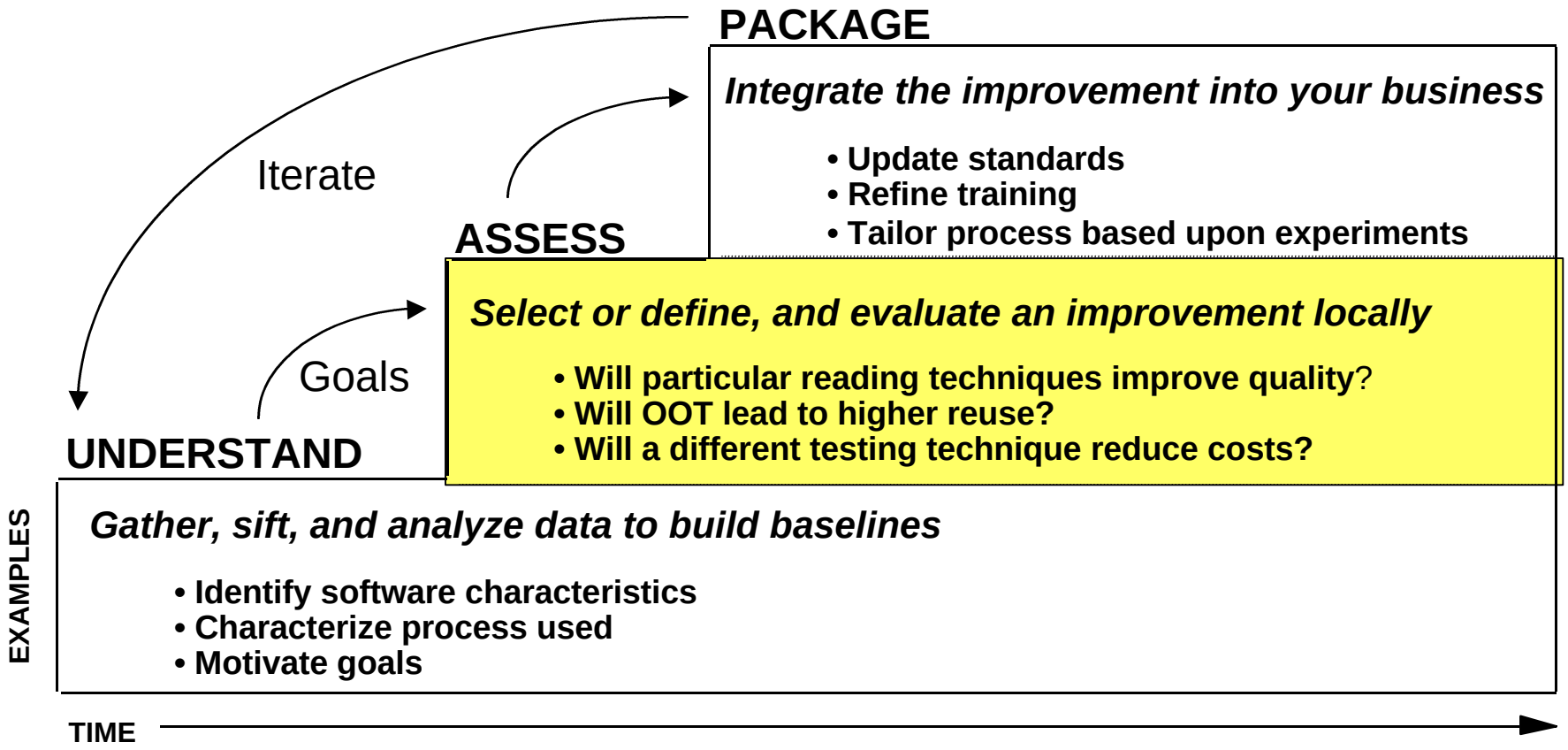
Evolving Knowledge in a Discipline

- Understanding a discipline involves learning, i.e.,
  - observation
  - reflection, and encapsulation of knowledge
  - model building (application domain, problem solving processes)
  - experimentation
  - model evolution over time
- This is the paradigm that has been used in many fields,
  - e.g., physics, medicine, manufacturing.
- The differences among the fields are
  - **how models are built and analyzed**
  - **how experimentation gets done**

Evolving Knowledge
In Software Engineering

- **Software engineering** is a **laboratory science**
- We need to understand the nature of the processes, products and the relationship between the two in the context of the system
- All software is not the same
  - there are a large number of variables that cause differences
  - their effects need to be understood and studied
- Currently,
  - **insufficient set of models** to reason about the discipline
  - **lack of recognition of the limits** of technologies for the context
  - there is **insufficient analysis and experimentation**
- This talk is about experimentation in the software discipline

Where Experiments/Knowledge Building fits in the Quality Improvement Paradigm



Evolving Bodies of Knowledge from Experiments

- Many categories: from controlled experiments to case studies
- Performed for many purposes: to study process effects, product characteristics, environmental constraints (cost or schedule).
- Typically they are looking for a relationship between two variables, such as the relationship between process characteristics and product characteristics
- **Problems** with experiments (controlled)
  - the **large number of variables** that cause differences
  - deal with **low level issues**, **microcosm of reality**, **small set of variables**
- **=> Combining experiments** is necessary to build a body of knowledge that is useful to the discipline

Criteria for building comprehensive bodies of knowledge in Software Engineering

- Sets of **high level hypotheses**
  - address interest of the software engineering community
  - identify sets of dependent and independent variables
  - provide options for the selecting detailed hypotheses
- Sets of **detailed hypotheses**
  - written in a context that allow for a well defined experiment
  - combinable to support high level hypotheses
- **Context variables** that can be changed to allow for
  - experimental design variation (make up for validity threats)
  - specifics of the process context;
- **Sufficient documentation** for replication and combination
- **Community of researchers** willing to collaborate and replicate.

Choosing a High Level Focus

- General Interest to the community
  - **Analyzing the Effects of a SE Process on a Product**
- What are the high level **questions of interest**?
  - Can we effectively design and study techniques that are procedurally defined, document and notation specific, goal driven, and empirically validated for use?
  - Can we demonstrate that a procedural approach to a software engineering task could be more effective than a less procedural one under certain conditions?
- What are the **high level hypotheses**?
  - A reading technique that is procedurally defined, document and notation specific, and goal driven for use is more effective than one that does not have these characteristics
  - A procedural approach to reading based upon specific goals will find different defects than one based upon different goals

Example: Understanding for Use
Motivation for Reading

Why pick reading?

- Reading is a **key technical activity** for analyzing and constructing software documents and products
- Reading is **a model for writing**
- Reading is **critical for reviews, maintenance, reuse, ...**

What is a reading technique?

- a concrete set of instructions given to the reader saying how to read and what to look for in a software product

More Specifically, software reading is

- **the individual analysis of a software artifact**
  - e.g., requirements, design, code, test plans
- **to achieve the understanding needed for a particular task**
  - e.g., defect detection, reuse, maintenance

Choosing a High Level Focus

- How do we build a framework for combining hypotheses from individual experiments, isolating out individual variables?
- Consider using the **Goal/Question/Metrics Paradigm**
- Goal Template:
  - Analyze an **object of study** in order to **purpose** with respect to **focus** from the point of view of **who** in the context of **environment**
- Consider decomposing each of the variables to identify and classify the independent, dependent, and context variables

Choosing a High Level Focus

- Analyzing the Effects of SE Processes on Products
  - Analyze **processes** to evaluate their effectiveness on a product from the point of view of the knowledge builder in the context of (variable set)

- Characterize the object of study:
  - Object of Study (**Process**, Product, …)
  - Process Class (Life Cycle Model, Method, **Technique**, Tool, …)
  - Technique Class (**Reading**, Testing, Designing, …)

- Analyze **reading techniques** to evaluate their effectiveness on a product from the point of view of the knowledge builder in the context of variable set

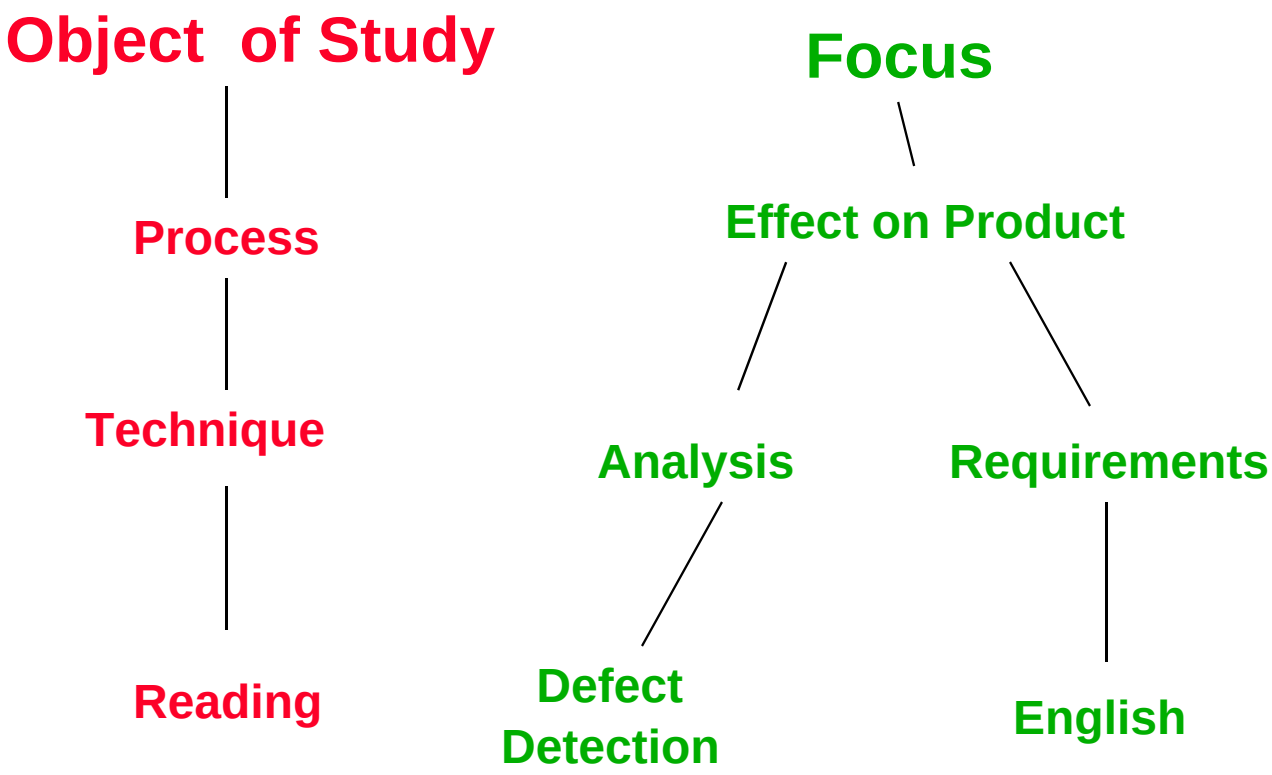
Choosing a High Level Focus

- Analyze reading techniques to evaluate their **effectiveness on products** from the point of view of the knowledge builder in the context of variable set (G1)

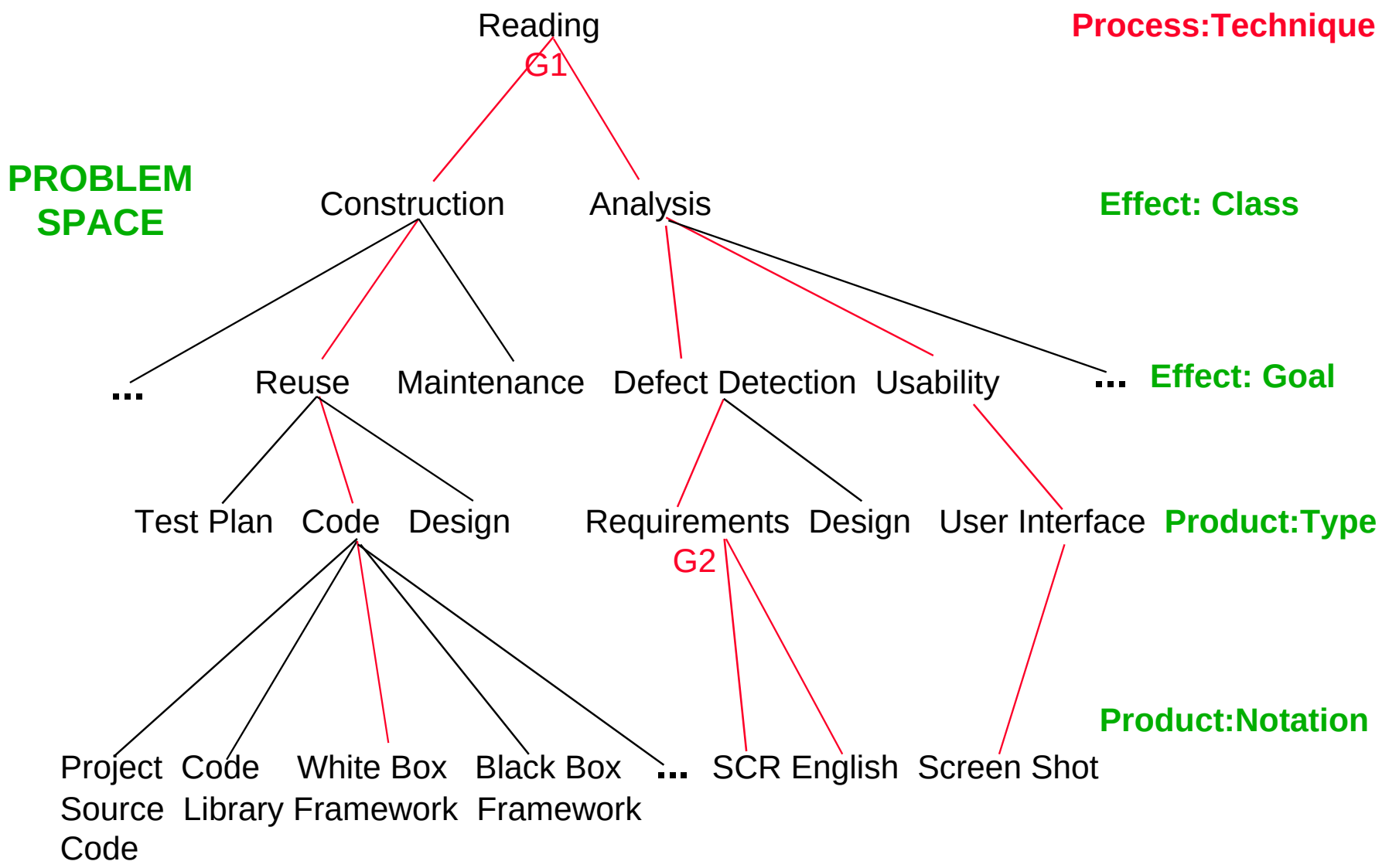
- Characterize the focus: **Effectiveness on a Product**
  - Effectiveness Class (Construction, **Analysis**, …)
  - Effectiveness Goal (**Defect Detection**, **Usability**, …
  - Product Type (**Requirements**, Design, Test Plan, **User Interface**, …
  - Product Notation (**English**, **SCR**, Mathematics, **Screen Shot**, …

- Example Goal: Analyze reading techniques to evaluate their **ability to detect defects in a Requirements Document** from the point of view of the knowledge builder in the context of variable set (G2)

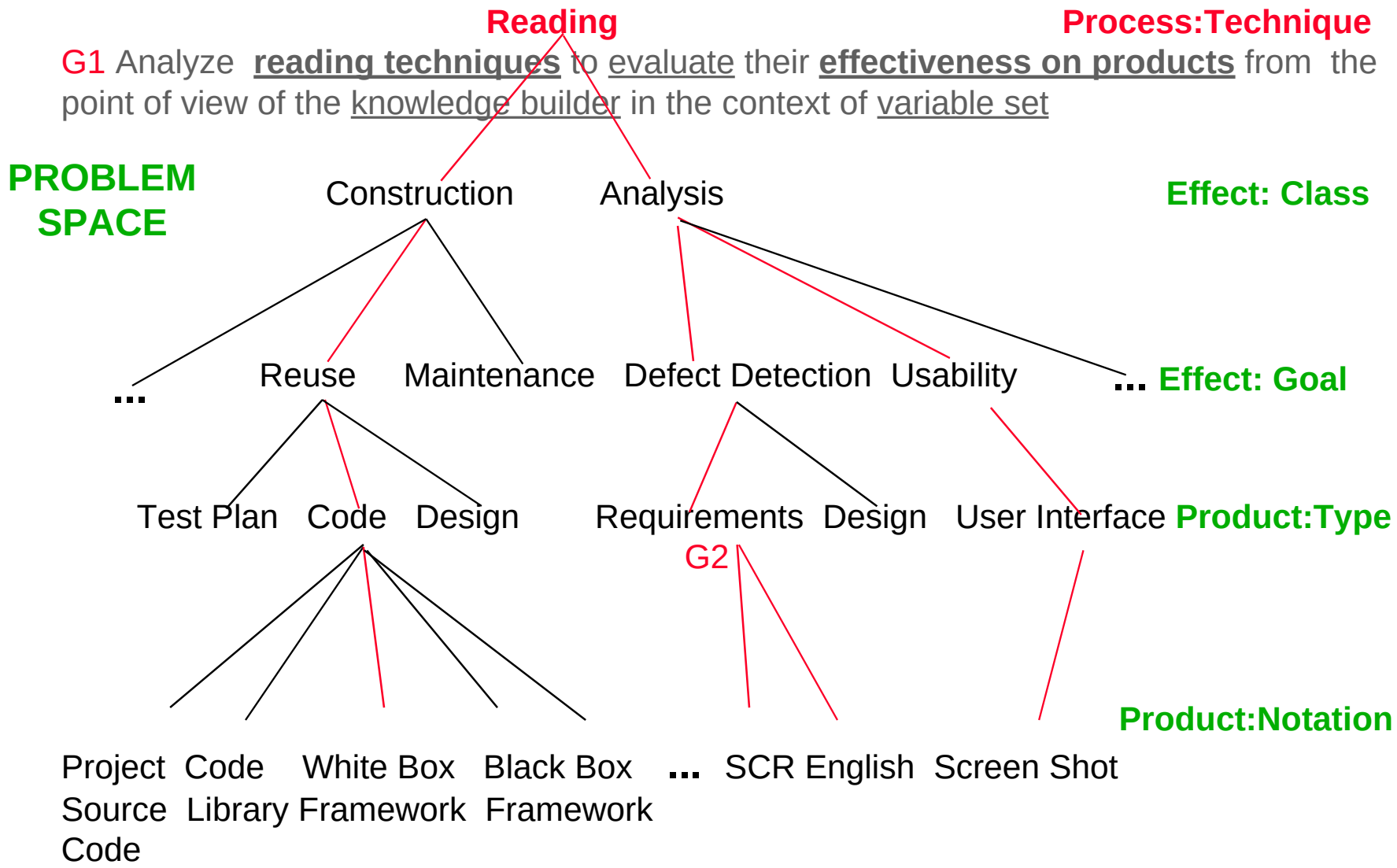
Refining a High Level Focus



Families of Reading Techniques



Families of Reading Techniques



Scenario-Based Reading Definition

- Given this set of characteristics/dimensions, an approach to generating a family of reading techniques, called **operational scenarios**, has been defined

- **Goals:** To define a set of reading technologies that can be
  - document and notation specific
  - tailorable to the project and environment
  - procedurally defined
  - goal driven
  - focused to provide a particular coverage of the document
  - empirically verified to be effective for its use
  - usable in existing methods, such as inspections

- These goals defines a set of guidelines/characteristics for a process definition for reading techniques that can be studied experimentally

Choosing a Specific Focus from the Experimental Framework

- Characterize the process:
  - Technique Class (**Reading**, Testing, Designing, …)
  - Technique Characteristics (**goal oriented**, **procedurally based**, **coverage focussed**, **documentation and notation specific**, …)

- Analyze a **set of goal-oriented, procedurally-based, coverage focussed, document and notation specific reading techniques** to evaluate their effectiveness on a product from the point of view of the knowledge builder in the context of (variable set)

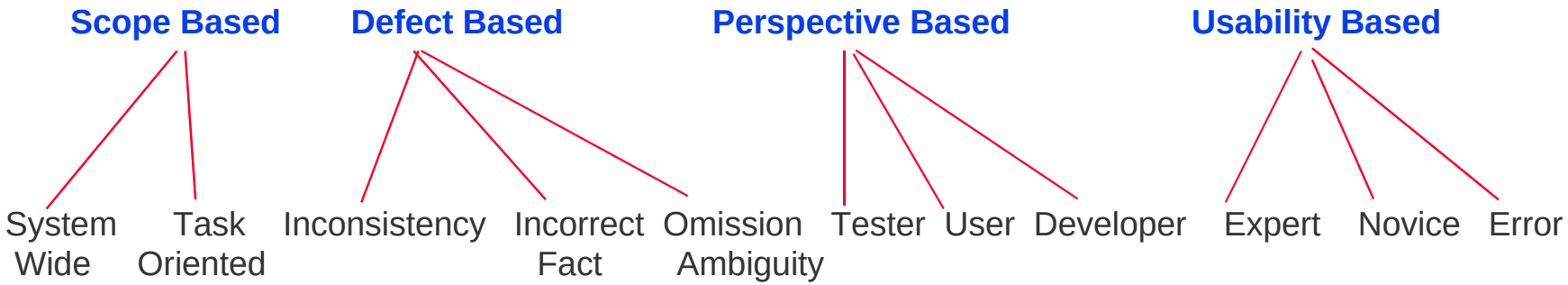
- Analyze a set of scenario based reading techniques to evaluate their effectiveness on products from the point of view of the knowledge builder in the context of (variable set)

- Attempts to satisfy the high level hypotheses and provide a frameworks for individual experiments

Choosing a Specific Focus from the Experimental Framework

- Analyze a set of scenario based reading techniques to evaluate their effectiveness on products from the point of view of the knowledge builder in the context of (variable set)

- We have developed four families of reading techniques
  - parameterized for use in different contexts and
  - evaluated experimentally in those contexts



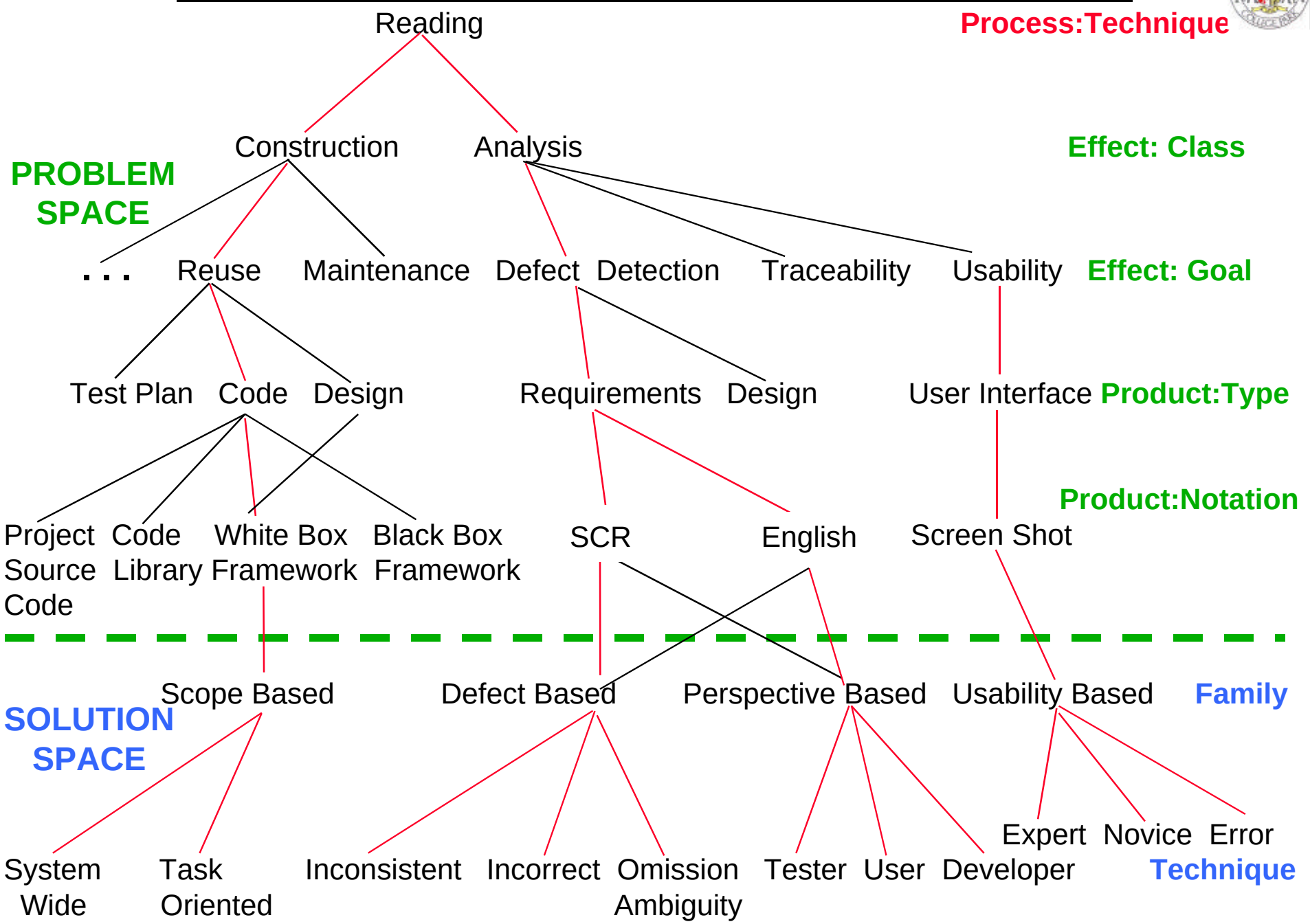
Choosing a Specific Focus from the Experimental Framework

- Analyze a **set of scenario based** **reading techniques** to evaluate their **ability to detect defects in a Requirements Document** from the point of view of the knowledge builder in the context of (variable set)

- Example: Perspective -Based Reading:
  - Choose perspectives; designer, tester, user
  - Define procedural processes for each perspective
  - Choose experimental treatment
  - Choose defect classes
  - etc.

- Contexts (context variables) can be continually expanded, e.g., NASA/SEL subjects, Professional Software Engineering student, Bosch project personnel

Families of Reading Techniques



Sample Set of Experiments

- We have run several experiments
  - on all four families of reading techniques
  - parameterized for use in different contexts
  - some involved us as directly as experimenters, others did not

- Example Contexts: (Government, University, Industry)
  - NASA/GSFC (PBR)
  - UM Professional SE Course (PBR, UBR)
  - UM Students (DBR, UBR, SBR)
  - Bureau of Census (UBR)
  - Robert Bosch (PBR)
  - Lucent (DBR)

- Example Countries: (U.S., Germany, Italy, Sweden, Scotland, Norway,...)

Choosing a Specific Focus from the Experimental Framework

- There are still many questions that need to be covered:
  - Process variable (**Independent variable**) issues:
    - How do we define/specify the process?
    - How do we account for process conformance?
  - Effectiveness of Product (**Dependent variable**) issues:
    - How do we select good criteria for effectiveness?
  - **Context Variables** Issues:
    - What subjects are performing the process?

- Questions associated with the variables need to be further specified and documented for replication

- Varying the values of these variables allow us to
  - vary the detailed hypotheses
  - support validity of study results

Designing Detailed Experiments to Increase Knowledge

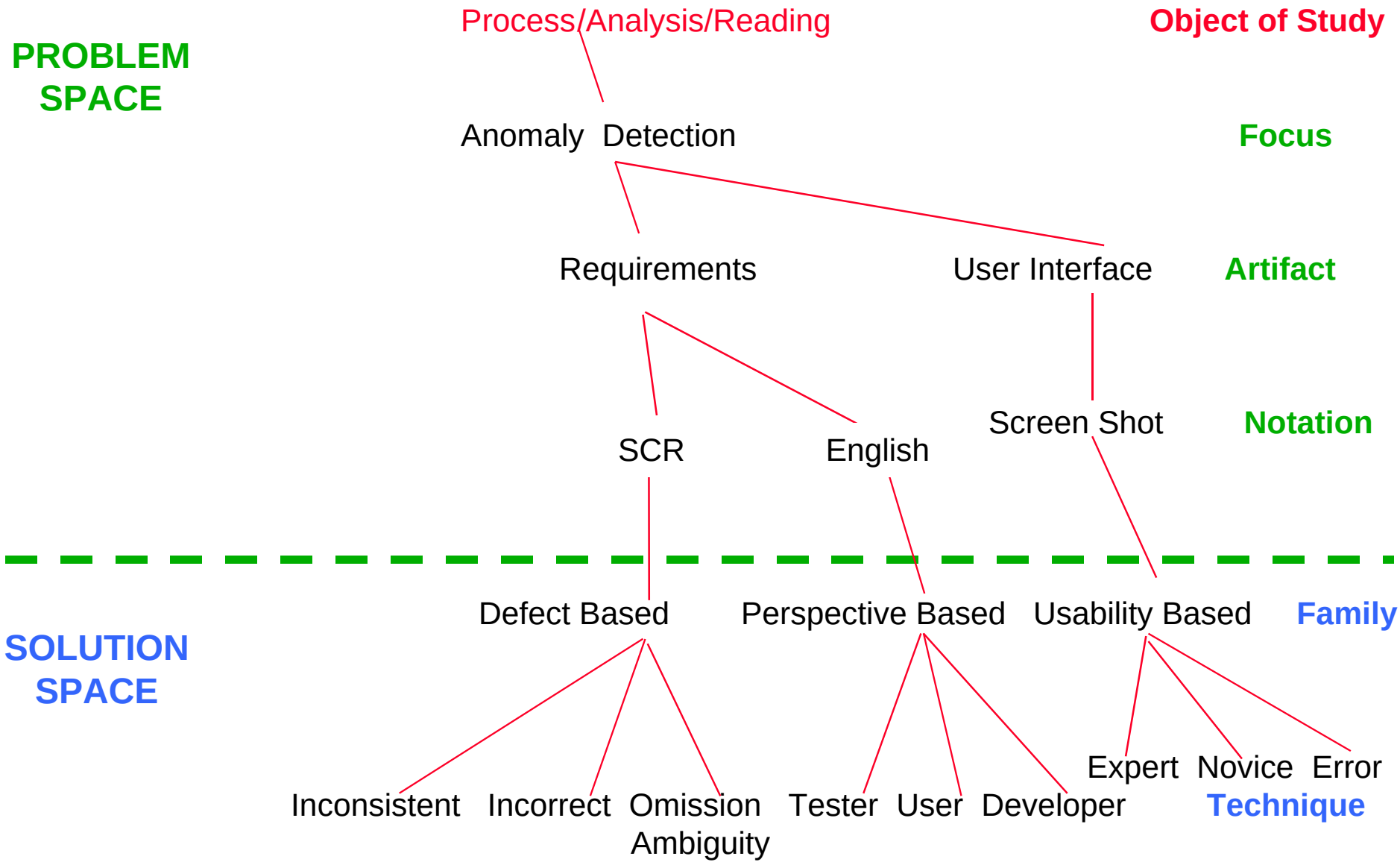
- We can build up knowledge by **replicating** detailed experiments, keeping the same hypothesis, combining results

- Varying **Context Variables**
  - subject experience
  - context (classroom, toy, off-line, in project)
  - variability among subjects
  - Vary order of events and activities

- Allows us to balance threats to validity
  - interaction of experience and treatment
  - spontaneous migration of subjects across treatments
  - replicating to counterbalance

Focused Families of Analysis Techniques

G3 Analyze a **set of processes focused to provide a particular coverage of an artifact** to evaluate their **ability to detect anomalies** from the point of view of the knowledge builder in the context of (variable set)



Conclusions from Experiments

- Able to **combine** the **results** of several experiments and **build up** our **knowledge** about software processes
  - We can **effectively design and study techniques** that are procedurally defined, document and notation specific, goal driven, and empirically validated for use
  - We can demonstrate that a **procedural approach** to a software engineering task could be more effective than a less procedural one under certain conditions (e.g., depends on experience)
  - A procedural approach to reading based upon **specific goals** will find defects related to those goals, so reading can tailored to the environment
  - et. al.

Conclusions about Knowledge Building Experimental Framework

- Benefit to Researchers:
  - ability to **increase the effectiveness** of individual experiments
  - offers a **framework** for building relevant practical SE knowledge
  - provides a way to develop and integrate **laboratory manuals**
  - generate a **community** of experimenters

- Benefits to Practitioners:
  - offers some relevant **practical SE knowledge**
  - provides a better basis for making judgements about **selecting process**
  - shows importance of and ability to tailor "**best practices**"
  - provides support for defining and **documenting processes**
  - allows organizations to **integrate their experiences** with processes

Contributors to This Work

- Directly to the Ideas Presented here:
  - Forrest Shull, Filippo Lanubile
- As Experimenters Locally:
  - Reported Experiments: Scott Green, Oliver Laitenberger, Filippo Lanubile, Forrest Shull, Sivert Sorumgaard, Marvin Zelkowitz, Zhijun Zhang
  - New Studies Underway: Fred Fredericks, Shari Lawrence Pfleeger, Rae Kwon, Guilherme Travassos
- As Experimenters in Other Locations
  - ISERN members
  - Others